All levels of SPASEQL are configurable. Implementers can choose the subset of features that works for their service. Below we list the complete set of features available.

SPASEQL Level 0

What a query can ask to be returned

- Time Spans
- Time
- Numerical Data Granule URL
- Display Data Granule URL
- Harvesting (shorthand for saying return new/modified metadata)

What a query can constrain on

- Release Date (for harvesting)
- ResourceID
- Cadence
- AssociationID
- InputResourceID
- Position
- ObservedRegion
- ObservatoryRegion
- Statistical Estimators
- Measurement Type
- Instrument ID
- Observatory ID
- Parameter (Quantity, Qualifier, Particle Type)
- Time Span
- Phenomena Type
- VSO capabilites

SPASEQL Level 1

Query constraints and return options are the same as in level 0. This level adds the capability for complex queries, i.e. [A and (B or C)]

SPASEQL Level 2

This level extends level 1 with the ability to constrain on any SPASE term. It is a superset of level 1 and as such contains all the capabilities found in levels 0 and 1.

Implicit in Query Language

Several assumptions are build into the query language and should be taken into account when implemented

- The logical operators and/or are optional in levels 1 and 2. The user is responsible for determining when these operators are needed. Any queries that are missing an operator should assume “and” and provide a warning message in the response.
- TimeSpan term is meant to indicate a time period that must be contained in the results. That is, the results do not have to be entirely contained within the given time span. For example, if a user asks for a time span of one day and a VxO provides monthly granules, then the granule containing the user specified day should be considered a match.
- Harvesting may return large amounts of metadata. In order to not overwhelm our systems harvesting requests should return links (URLs) to new/modified metadata not the actual metadata.
- In level 2 the format for “Term” is SPASE_Resource_Type::Term. For example, NumericalData::Description. Matches are at the first occurance of the term. For example, NumericalData::ResourceHeader::URL would match the Information URL while NumericalData::AccessInformation::URL would match the Access URL.